(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平10－154116

(43)公開日　平成10年(1998)６月９日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　13/00	３５３	Ｇ０６Ｆ　13/00	３５３Ｓ
	３５１		３５１Ｅ

審査請求　未請求　請求項の数９　ＯＬ　（全 18 頁）

(21)出願番号　特願平8－313805

(22)出願日　平成８年(1996)11月25日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　山▲崎▼　康雄
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内

(72)発明者　森　利明
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内

(72)発明者　鵜飼　敏之
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内

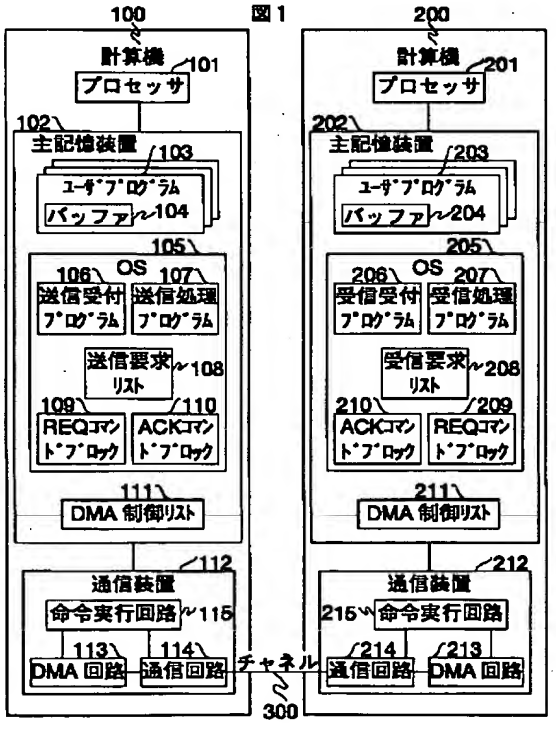
(74)代理人　弁理士　高橋　明夫

最終頁に続く

(54)【発明の名称】　データ転送方法

(57)【要約】
【課題】ＯＳにより送信データをバッファリングしないで転送する。
【解決手段】送信側の計算機１００では、ＯＳ１０５は、複数のユーザプログラム１０３から指定された複数の送信データの送信先ユーザプログラム２０３がそれぞれデータを受信可能な状態にあるかをＯＳ２０５と通信してまとめて確認する。受信側のユーザプログラム２０４が受信要求を発行していない場合には、そのプログラムは受信不可とし、送信可と判断された複数の送信データの送信をまとめて通信装置１１２に指示する。通信装置１１２では、ＤＭＡ回路１１３が各ユーザプログラム１０３のバッファ１０４から送信データを直接連続して読み出し、通信回路１１４がそれぞれの送信データをチャネル３００を介して送信する。受信側の計算機２００では、ＤＭＡ回路２１３が各送信データを送信先のユーザプログラム１０４のバッファ２０４に直接連続して書き込む。

図１



【特許請求の範囲】

【請求項１】（ａ）第１の計算機の送信元のユーザプログラムにより、該第１の計算機の主記憶装置上のその送信元のユーザプログラム用の第１の領域に保持されたデータを、その第１の計算機に伝送路を介して接続された第２の計算機の送信先のユーザプログラムへ送信することを要求する送信要求をその第１の計算機を制御する第１のＯＳに発行し、（ｂ）該送信要求に応答して、該第１のＯＳから該第２の計算機を制御する第２のＯＳに、該送信先のユーザプログラムがデータ受信可能な状態にあるか否かを問い合わせ、（ｃ）該問い合わせに対して、該第２のＯＳが該送信先のユーザプログラムがデータ受信可能な状態にあるか否かを判断し、その判別結果を示す問い合わせ結果を該第２のＯＳにより該第１のＯＳに通知し、（ｄ）該第２のＯＳが該送信先のユーザプログラムがデータ受信可能な状態にあると判断したときに、該第２のＯＳにより、該送信要求が指定したデータの受信と、該第２の計算機の主記憶装置上に定められた該送信先のユーザプログラム用の第２の領域へのそのデータの書き込みとを該第２の計算機の通信装置に指示し、（ｅ）該問い合わせ結果が該送信先のユーザプログラムがデータ受信可能な状態にあることを示すとき、該第１のＯＳにより上記データの送信を該第１の計算機の通信装置に指示し、（ｆ）該第１のＯＳによる該指示に応答して、該第１の計算機の該通信装置により該データを該第１の領域から読み出し、該第２の計算機に該伝送路を介して転送し、（ｇ）該第２のＯＳによる該指示に応答して、該第２の計算機の該通信装置により、該データを受信し、該第２の領域に書き込むステップからなるデータ転送方法。

【請求項２】該問い合わせ結果が、該送信先のユーザプログラムがデータ受信不可能な状態にあることを示すとき、該第１のＯＳにより、該ステップ（ｂ）を繰り返すステップをさらに有する請求項１記載のデータ転送方法。

【請求項３】該送信先のユーザプログラムから該データの受信要求を該第２のＯＳに要求するステップをさらに有し、

該ステップ（ｃ）は、

該問い合わせを受けた時点で、該送信先のユーザプログラムから該データの受信要求がすでに発行されているか否かに応じて、該送信先のユーザプログラムが受信可能な状態にあるか否かを判別するステップを有する請求項１または２記載のデータ転送方法。

【請求項４】上記送信要求は、上記データのサイズを指定し、

上記受信要求は、該送信先のユーザプログラム用の上記領域のサイズを指定し、

ステップ（ｃ）は、

該問い合わせを受けた時点で、該送信先のユーザプログラムから該データの受信要求がすでに発行されているときに、上記データのサイズが該第２の領域のサイズを越えないか否かをさらに判別し、

該問い合わせを受けた時点で、該送信先のユーザプログラムから該データの受信要求がすでに発行されているときでも、上記データの上記サイズが該第２の領域のサイズを超えているときには該送信先のユーザプログラムが受信不可能な状態にあると判別するステップを有する請求項３記載のデータ転送方法。

【請求項５】上記ステップ（ａ）は、該複数の送信元のユーザプログラムにより該第２の計算機で実行中の複数の送信先のユーザプログラムに対して複数の送信要求が発行されるように、該第１の計算機で実行中の複数の送信元のユーザプログラムの各々により実行され、

上記ステップ（ｂ）は、該複数の送信要求が指定する該複数の送信先のユーザプログラムがそれぞれデータ受信可能な状態にあるか否かを該第１のＯＳから該第２のＯＳにまとめて問い合わせるステップからなり、

上記ステップ（ｃ）は、該第２のＯＳが、該複数の送信先のユーザプログラムがそれぞれデータを受信可能な状態にあるか否かを判断し、それぞれの送信先のユーザプログラムに対する判別結果を示す複数の問い合わせ結果を該第２のＯＳにより該第１のＯＳにまとめて通知するステップからなり、

上記ステップ（ｄ）は、該第２のＯＳにより、該複数の送信先のユーザプログラムの内の一部の複数の送信先のユーザプログラムがデータを受信可能な状態にあると判断されたときに、該一部の複数の送信先のユーザプログラムに対する複数の送信要求が指定した一部の複数のデータを、該第２の計算機の該主記憶装置上に定められた該一部の該複数の送信先のユーザプログラム用の一部の複数の領域へそれぞれ書き込むことを該第２の計算機の該通信装置にまとめて指示するステップからなり、

上記ステップ（ｅ）は、該複数の問い合わせ結果に応答して、該第１のＯＳにより、該第１の計算機の該主記憶装置上に保持された該一部の複数の送信元のユーザプログラムが指定した一部の複数のデータの送信を該第１の計算機の該通信装置にまとめて指示するステップからなり、

上記ステップ（ｆ）は、該第１のＯＳによる該指示に応答して、該第１の計算機の該通信装置により、該一部の複数のデータを該第１の計算機の該主記憶装置から順次読み出し、該第２の計算機に該伝送路を介して順次転送するステップからなり、

上記ステップ（ｇ）は、該第２の計算機の該通信装置により、該一部の複数のデータを順次受信し、かつ、該第２の計算機の該主記憶装置上の該一部の複数の領域に書き込むステップからなる請求項１記載のデータ転送方法。

【請求項６】該複数の問い合わせ結果のいずれかが少な

くとも一つの他の送信元ユーザプログラムがデータ受信不可能な状態にあることを示すとき、該第１のＯＳにより、該少なくとも一つの他の送信元ユーザプログラムが発行した送信要求に関して該ステップ（ｂ）を繰り返すステップをさらに有する請求項５記載のデータ転送方法。

【請求項７】上記ステップ（ｂ）は、一定時間間隔で繰り返し実行される請求項５記載のデータ転送方法。

【請求項８】該複数の送信先のユーザプログラムのそれぞれからデータの受信要求を該第２のＯＳに要求するステップをさらに有し、

該ステップ（ｃ）は、

該問い合わせを受けた時点で、該複数の送信先のユーザプログラムの各々からデータの受信要求がすでに発行されているか否かに応じて、その送信先のユーザプログラムが受信可能な状態にあるか否かを判別するステップを有する請求項５または６記載のデータ転送方法。

【請求項９】上記複数の送信要求は、それぞれ送信すべきデータのサイズを指定し、

上記各受信要求は、該第２の計算機の該主記憶装置上に形成されたその受信要求を発行した送信先のユーザプログラム用の領域のサイズを指定し、

ステップ（ｃ）は、

該問い合わせを受けた時点で、各送信要求が要求する送信先のユーザプログラムからデータの受信要求がすでに発行されているときに、その送信要求が指定するデータのサイズがその受信要求が指定したその送信先プログラム用の上記領域のサイズを超えないか否かをさらに判別し、

該問い合わせを受けた時点で、各送信要求が指定する送信先のユーザプログラムからデータの受信要求がすでに発行されているときでも、その送信要求が指定する上記データの上記サイズがその送信先のユーザプログラム用の上記領域のサイズを超えているときにはその送信先のユーザプログラムが受信不可能な状態にあると判別するステップを有する請求項８記載のデータ転送方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は伝送路で接続された計算機間のデータ通信方法に係る。

【０００２】

【従来の技術】従来の計算機システム間でのデータ転送方法としては、ハイレベル・データリンク制御手順（ＨＤＬＣ：Ｈｉｇｈ　Ｌｅｖｅｌ　Ｄａｔａ　Ｌｉｎｋ　Ｃｏｎｔｒｏｌ）がある。例えば、“データ通信”、データ通信教育委員会編、共立出版、ｐｐ．４６－－５３（１９８２）に示す「ハイレベル・データリンク制御手順（ＨＤＬＣ：Ｈｉｇｈ　Ｌｅｖｅｌ　Ｄａｔａ　Ｌｉｎｋ　Ｃｏｎｔｒｏｌ）」参照。ＨＤＬＣは送信側の計算機が受信側の計算機に転送の通知を出し、受信側の計算機から応答が返ってきてからデータ転送を行う、というデータ転送方法である。このデータ転送方法は、直接データ転送路で接続された計算機間のデータ転送に適用できる。このＨＤＬＣ手順は、ユーザプログラム間でデータを直接転送することができるので高速転送に向いているが、それぞれＯＳで制御される任意のユーザプログラム間のデータ転送には適用できない。なお、ＨＤＬＣ手順によるデータ転送を高速化するために、送信側計算機の送信回路に、送信側のプログラムのバッファ内の送信データを連続して読み出すＤＭＡ回路を設け、受信側計算機の受信回路に受信したデータを受信側のプログラムのバッファに連続して書き込むＤＭＡ回路を設ける工夫もなされている。

【０００３】それぞれオペレーティングシステム（ＯＳ）で制御される任意のユーザプログラム間のデータ転送の方法としては、ＴＣＰ／ＩＰ（ＴｒａｎｓｍｉｓｓｉｏｎＣｏｎｔｒｏｌ　Ｐｒｏｔｏｃｏｌ／Ｉｎｔｅｒｎｅｔ　Ｐｒｏｔｏｃｏｌ）による方法がある。例えば、中村明／相田仁／計宇生／小池汎平共訳「ＵＮＩＸ４．３ＢＳＤの設計と実装」丸善、ｐｐ．１６９－１８５（１９９１）（原著：Ｔｈｅ　Ｄｅｓｉｇｎ　ａｎｄ　Ｉｍｐｌｅｍｅｎｔａｔｉｏｎ　ｏｆ　ｔｈｅ　４．３ＢＳＤ　Ｏｐｅｒａｔｉｎｇ　Ｓｙｓｔｅｍ、原著者：Ｓ．Ｊ．Ｌｅｆｆｌｅｒ／Ｍ．Ｋ．ＭｃＫｕｓｉｃｋ／Ｍ．Ｊ．Ｋａｒｅｌｓ／Ｊ．Ｓ．Ｑｕａｒｔｅｒｍａｎ、原発行所Ａｄｄｉｓｏｎ　Ｗｅｓｌｅｙ）参照。このデータ転送方法は、計算機ネットワークで接続された任意の計算機間での、他の計算機を介したパケットによるデータ転送にも使用できる。

【０００４】この方法では、複数のユーザプログラムが同時にデータを転送できるようにＯＳがバッファリングを行っている。すなわち、送信側の計算機では、ＯＳは、ユーザプログラムに割り当てられたバッファ領域からそのＯＳ内バッファへ送信すべきデータをコピーする。この際、コピーしたデータにパケットヘッダを付加する。この送信データとヘッダを含むパケットを送信回路によりネットワークを介して受信側計算機に送信する。受信側計算機では、受信回路が、受信したパケットを受信側ＯＳ内のバッファに格納し、受信側ＯＳは、書き込まれたパケットのヘッダから受信先のユーザプログラムを判別し、その受信側ユーザプログラムから受信要求が来たときに、その受信側のプログラムのバッファへ、受信側ＯＳのバッファ内にある受信したパケット内のデータをコピーする。この時点で受信側のＯＳは受信完了を送信側のＯＳに通知する。このデータ転送方法では、一つのパケットで送信されるデータの長さは、送信側ＯＳおよび受信側ＯＳ用のバッファのサイズに固定されている。このバッファの大きさを超えるデータを転送する場合にはこのバッファのサイズでもってそのデータを分割し、分割された複数の部分データを複数のパケッ

トにより転送する。送信側計算機のＯＳおよび受信側計算機のＯＳでは、通信装置の起動処理、終了処理をそれぞれのパケットに対して行う。

【０００５】この転送方法では、受信側のＯＳ内のバッファに、受信データを一時的に保持するので、受信側のユーザプログラムは、上記データの送信と非同期に受信要求を発行できる。また、もし、上記データの送信が不成功となったときには、送信側のＯＳがそのＯＳ用のバッファ内の送信データを再送する。このため、送信側のユーザプログラムは、受信側のユーザプログラムが受信可能な状態になっているか否かによらないでデータの送信を送信側のＯＳに要求でき、しかも、そのデータがそのＯＳ内のバッファにコピーされた後は、そのデータの転送が完了するのを待たないで他の処理を実行できる。

【０００６】この方法によるデータ転送を高速に行うために、送信側計算機の送信装置内に、送信側のＯＳのバッファから送信データを連続して読み出すためのＤＭＡ回路を設け、受信側計算機の受信装置内に、受信したデータを受信側ＯＳのバッファに連続して書き込むためのＤＭＡ回路を設けるという工夫も行われている。例えば、山居正幸著「イーサネット・ボード＆カード」、オープンデザイン、Ｎｏ．３、ＣＱ出版、ｐｐ．１０９－－１１３（１９９４）参照。

【０００７】

【発明が解決しようとする課題】ＴＣＰ／ＩＰによるデータ転送では、送信側の計算機および受信側の計算機のいずれにおいてもユーザプログラムのバッファとＯＳのバッファの間でデータコピーが行われる。このデータコピー処理にかかる時間はデータ量に比例して大きくなり、このデータコピー時間が、データ転送時間に占める割合が大となる。その結果、計算機間を高速な伝送路で接続してもこの方法によるデータ転送速度はさほど改善されない問題点がある。前述のように、送信側計算機の送信回路と受信側計算機の受信回路の両方にＤＭＡ回路を設けてもこの問題は残る。とくに科学計算機分野では、非常に多量のデータを計算機間で連続して転送する必要が生じ、このデータコピー時間の問題が著しく大となる。

【０００８】さらに、この方法では、送信側ＯＳおよび受信側ＯＳ用のバッファのサイズを超えるデータを転送する場合、前述のように、そのデータを分割して得られる複数の部分データを複数のパケットにより転送する。この際、送信側計算機のＯＳおよび受信側計算機のＯＳでは、通信装置の起動処理、終了処理をそれぞれのパケットに対して行う。このデータのサイズが大きくなると、それを転送するのに必要なパケット数が多くなり、これらのＯＳが行う信装置の起動処理、終了処理に時間が掛かり、それによりデータ転送速度が低減するという問題がある。この問題も科学計算機分野のごとく非常に多量のデータを計算機間で連続して転送する場合に大きくなる。

【０００９】ＴＣＰ／ＩＰによるデータ転送では、多数の計算機を経由したデータ転送にも使用できるが、その経由のためにデータ転送速度は低下する。科学技術計算用のシステムでは、伝送路で結合された特定の計算機間で他の計算機を介さないでより高速にデータを転送することが望まれる。

【００１０】従って、本発明の目的は、送信側計算機のＯＳで制御されるユーザプログラムから、受信側計算機内のＯＳで制御されるユーザプログラムに、より高速にデータを転送するデータ転送方法、とくに、科学計算用のユーザプログラムの間で非常に多量のデータを他の計算機を介しないで転送するのに適したデータ転送方法を提供することである。

【００１１】本発明のより具体的な目的は、送信側計算機でのユーザプログラム用のバッファとＯＳ用のバッファ間のデータコピーおよび受信側計算機でのユーザプログラム用のバッファとＯＳ用のバッファ間のデータコピーを行わないで、それらのユーザプログラム間でデータを転送するデータ転送方法を提供することにある。

【００１２】本発明の他の具体的な目的は、送信側計算機のユーザプログラムが指定したデータを分割しないで受信計算機のユーザプログラムに転送するデータ転送方法を提供することにある。

【００１３】

【課題を解決するための手段】上記目的を達成するために、本発明によるデータ転送は、相互に伝送路で接続された第１、第２の計算機を有し、該第１、第２の計算機がプロセッサと主記憶装置と該主記憶装置上に格納されたデータを読み出し、該伝送路に送信しあるいは該伝送路からデータを受信し、該主記憶装置に書き込む通信装置を有するシステムにおいて実行される。まず、第１の計算機で実行されている送信元のユーザプログラムと第２の計算機で実行されている送信先のユーザプログラムの間でデータ転送が可能か否かを第１の計算機を制御する第１のＯＳと第２の計算機を制御する第２のＯＳとの間の通信により判別する。もしそれらのプログラム間でデータ転送が可能と判断されたときには、それらのＯＳからの指示に従って、送信側計算機の通信装置と受信側計算機の通信装置によりそれらのユーザプログラムの間でデータをそれらのＯＳを介しないで直接転送する。より具体的には、（ａ）送信元のユーザプログラムにより、その送信元のユーザプログラム用の第１の領域に保持されたデータを第２の計算機で実行されている送信先のユーザプログラムへ送信することを要求する送信要求を第１のＯＳに発行し、（ｂ）第１のＯＳから第２のＯＳに、送信先のユーザプログラムがデータ受信可能な状態にあるか否かを問い合わせ、（ｃ）第２のＯＳが送信先のユーザプログラムがデータ受信可能な状態にあるか否かを判断し、その判別結果を示す問い合わせ結果を第

２のＯＳにより第１のＯＳに通知し、（ｄ）第２のＯＳが送信先のユーザプログラムがデータ受信可能な状態にあると判断したときに、第２のＯＳにより、送信要求が指定したデータの受信と、送信先のユーザプログラム用の第２の領域へのそのデータの書き込みとを第２の計算機の通信装置に指示し、（ｅ）問い合わせ結果が送信先のユーザプログラムがデータ受信可能な状態にあることを示すとき、送信側計算機のＯＳにより上記データの送信を送信側計算機の通信装置に指示し、（ｆ）第１の計算機の通信装置によりデータを送信元ユーザプログラム用の第１の領域から読み出し、第２の計算機に伝送路を介して転送し、（ｇ）第２の計算機の該通信装置により、データを受信し、送信先のユーザプログラム用の上記第２の領域に書き込む。

【００１４】

【発明の実施の形態】以下、本発明に係るデータ転送方法を図面に示した実施の形態を参照してさらに詳細に説明する。

【００１５】＜発明の実施の形態＞

（１）システム構成と動作の概要

図１に示した計算機システムにおいて、計算機１００、２００はそれぞれプロセッサ１０１、２０１と、主記憶装置１０２、２０２と、通信装置１１２、２１２とからなり、これらの通信装置が高速かつ高信頼な伝送路であるチャネル３００で接続され、これらの計算機は、このチャネル３００を介して他の計算機を介することなく直接に接続されている。通信装置１１２、２１２は、それぞれダイレクトメモリアクセス（ＤＭＡ）回路１１３、２１３および通信回路１１４、２１４、命令実行回路１１５、２１５からなり、ＤＭＡ回路１１３、２１３はそれぞれ主記憶装置１０２、２０２上の領域を直接読み書き可能になっている。ＯＳ１０５、２０５が通信装置１１２、２１２への命令を主記憶装置１０２、２０２上に用意されたＤＭＡ制御リスト１１１、２１１に格納して通信装置１１２、２１２を起動すると、命令実行回路１１５、２１５がＤＭＡ制御リスト１１１、２１１に格納された命令を解釈し、ＤＭＡ回路１１３、２１３および通信回路１１４、２１４を制御することで与えられた命令を実行し、複数の命令がある場合は残りの命令も同様に順次実行し、全ての命令の解釈し、その実行が完了すると、ＯＳ１０５、２０５に対してハードウェア割り込みを発生して命令の完了報告を行う。ＤＭＡ制御リスト１１１、２１１に格納する命令は、（１）主記憶装置１１２、２１２のどこから、（２）どれだけの大きさを、（３）送信するかあるいは受信するか、という指定の組からなる任意の個数のＤＭＡ制御命令のリストである。通信装置１１２、２１２の双方で対となるＤＭＡ制御命令の組が実行されると通信路３００を介した通信が成立する。対となるＤＭＡ制御命令の組とは、同一の大きさの送信ＤＭＡ制御命令と受信ＤＭＡ制御命令の組である。例えば計算機１００において命令実行回路１１５が「主記憶装置１１２のアドレスＡより、大きさａだけ送信」という命令を解釈すると、ＤＭＡ回路１１３に対しては指定された領域（アドレスＡより大きさａ）を読み出してその結果を通信回路１１２に渡すように指示し、通信回路１１２に対しては渡されたデータを伝送路３００に出力するように指示する。その場合、計算機２００において命令実行回路２１５に与えられる命令は「主記憶装置２１２のアドレスＢより、大きさａだけ受信」でなければならない。そして、通信回路１１２の送信動作と通信回路２１２の受信動作の双方が行われた時点より伝送路３００を介した通信が開始される。通信回路１１２、２１２は、送信動作あるいは受信動作のどちらかいずれかが先に発生した場合は、受信動作あるいは送信動作が発生するのを待つ。

【００１６】主記憶装置１０２、２０２において、計算機１００、２００を制御するＯＳ１０５、２０５の制御下で複数のユーザプログラム１０３、２０３が動作している。各々のユーザプログラム１０３、２０３には各計算機内で一意なプログラム識別子が割り当てられており、ＯＳ１０５、２０５はこのプログラム識別子で指定して、ユーザプログラムの動作を中断し、あるいは再開することができる。

【００１７】複数のユーザプログラム１０３の各々は、送信データをバッファ１０４に格納した後に、ＯＳ１０５に送信要求を発行する。ＯＳ１０５は、各ユーザプログラム１０３内のバッファ１０４に格納された送信データを受信側計算機２００に転送することを通信装置１１２に指示する。通信装置１１２では、従来技術と異なり、ＤＭＡ回路１１３が各ユーザプログラム１０３内のバッファ１０４から送信データを直接連続して読み出す。ＯＳ１０５は、各ユーザプログラム１０３が指示した送信データの送信を通信装置１１２に指示するが、従来技術と異なり、その送信データをそのＯＳ１０５内のバッファにコピーしない。通信回路１１４が、ＤＭＡ回路１１３が読み出した送信データをチャネル３００を介して受信側の計算機２００に送信する。

【００１８】受信側の計算機２００の通信装置２１２では、通信装置２１４は、それぞれの送信データを受信するごとに、従来技術と異なり、ＤＭＡ回路２１３が各送信データの送信先のユーザプログラム１０４内のバッファ２０４に受信されたデータを、受信側計算機を制御するＯＳ２０２を介することなく直接連続して書き込む。ＯＳ２０５は、従来技術と異なり、受信データをそのＯＳ２０５内のバッファにコピーしない。

【００１９】上記の方法によりデータを転送するには、受信側のユーザプログラム２０３と上記データ転送との同期を取る必要がある。すなわち、複数のユーザプログラム１０３のそれぞれから指定された送信データの送信先のユーザプログラム２０３がデータを受信可能な状態

になければならない。ユーザプログラム２０３が受信可能な状態とは受信したデータを蓄えるためのバッファ２０４が用意されているということである。このバッファ２０４を繰り返し使用する場合などにバッファ２０４の上書きによるデータ破壊を防止するため、受信ユーザプログラム２０３がＯＳ２０５に対して明示的に受信要求を発行することによって、受信準備が整ったことすなわち受信可能な状態であることを通知する。受信要求の発行は後述する受信受付プログラム２０６の呼び出しで行う。このために、本実施の形態では、ＯＳ１０５は、複数のユーザプログラム１０３から指定された送信データの送信先のユーザプログラム２０３がデータを受信可能な状態にあるかをＯＳ２０５と通信してまとめて確認し、この確認後にそれらのユーザプログラム１０３から指定された複数の送信データの送信をまとめて通信装置１１２に指示する。

【００２０】具体的には、ＯＳ１０５内の送信受付プログラム１０６が、複数のユーザプログラム１０３から発行された複数の送信要求を送信要求リスト１０８に蓄え、送信処理プログラム１０７が一定の時間間隔ごとに、それらの蓄えられた送信要求を表すＲＥＱコマンドブロック１０９を作成し、さらに、ＤＭＡ制御リスト１１１を生成して通信装置１１２にＲＥＱコマンドブロック１０９の送信を指示する。通信装置１１２が、このブロック１０９を計算機２００に送信すると、計算機２００では、いずれかのユーザプログラム２０３が計算機１００内のいずれかのユーザプログラム１０３からデータを受信すべき状態になると、データ受信要求をＯＳ２０５に発行するように構成され、ＯＳ２０５内の受信受付プログラム２０６は、この受信要求を受信要求リスト２０８に格納する。

【００２１】通信回路２１４がこのＲＥＱコマンドブロック１０９を受信すると、ＤＭＡ回路２１３が主記憶装置２０２にＲＥＱコマンドブロック２０９として書き込み、受信処理プログラム２０７は、受信要求リスト２０８に保持された受信要求と書き込まれたＲＥＱコマンドブロック２０９とに基づいて、このＲＥＱコマンドブロック２０９が指定する送信先のユーザプログラムが受信可能な状態にあるか否かをそれぞれ示すＡＣＫコマンドあるいはＮＡＣＫコマンドを含むＡＣＫコマンドブロック２１０を作成し、さらに、ＤＭＡ制御リスト２１１を生成して通信装置２１２にそのＡＣＫコマンドブロック２１０の送信を指示する。計算機１００では、ＡＣＫコマンドブロック２１０を受信すると、送信処理プログラム１０７が、このＡＣＫコマンドブロック２１０に基づいて、先に送信したＲＥＱコマンドブロック１０９内の送信要求の内、送信可能なもののみに付随するデータの送信を通信装置１１２に指示する。

【００２２】なお、ＯＳ１０５、２０５は同じ構造を有し、いずれの計算機も他方の計算機に対して送信および受信を行える。従って、計算機１００内のＯＳ１０５には、受信受付プログラム２０６、受信処理プログラム２０７、受信要求リスト２０８を有し、計算機２００内のＯＳ２０５には、送信受付プログラム１０６、送信処理プログラム１０７、送信要求リスト１０８を有する。しかし、図では簡単化のために、計算機１００が送信側の計算機、計算機２００が受信側の計算機として動作する場合に必要な部分のみを図示してある。以下、本実施の形態におけるデータ転送方法の詳細を、図２に示す主要部分の動作のタイムチャートを参照して説明する。なお、以下では、バッファ１０４と２０４、ＲＥＱコマンドブロック１０９、２０９、ＡＣＫコマンドブロック２１０、１１０とＤＭＡ制御リスト１１１、２１１は主記憶装置１０２あるいは２０２上で連続して常駐しているものと仮定する。領域が「連続する」とは仮想的な記憶上ではなく物理的な主記憶上において連続して領域が確保されているという意味であり、領域が「常駐する」とは通常のユーザプログラムの扱う記憶領域とは異なり確保された領域が必要に応じてＯＳ１０５、２０５によりディスク装置に待避されることがなく常に物理的な主記憶上に存在し続けるということを意味する。これは通信装置１１２、２１２からは主記憶装置１０２、２０２が仮想的な領域として認識できないためである。また、全てのコマンドブロック１０９、２１０、１１０、２０９は等しい大きさであると仮定する。転送するファイルの数やそれぞれのファイルの大きさなどについての情報は、送信側と受信側のユーザプログラム間で予め明らかであるとする。

【００２３】（２）データ転送動作の詳細

（Ａ）送信側計算機１００（その１）ＲＥＱコマンドの送信

（Ａ１）送信側のユーザプログラム１０３からの送信要求の発行

送信側の計算機１００において、三つの送信側ユーザプログラム１０３が走行し、受信側の計算機でも三つの受信側ユーザプログラム２０３が走行していると仮定する。

【００２４】図３のフローチャートを参照するに、三つの送信側ユーザプログラム１０３の各々の処理は、２重のループ構造となっており、処理の最初と最後に転送すべきファイルを保持するための送信用のバッファ１０４を確保し、あるいは解放する（ステップ４０１、４０８）。各ユーザプログラム用のバッファ１０４は、それが確保された後解放されるまでの間は主記憶装置１０２上に常駐し、全処理中に渡って繰り返し使用されるものとする。外側のループは複数のファイルを処理するためのループであり、内側のループは各々のファイルを複数回の通信で転送するためのループである。

【００２５】各送信側ユーザプログラム１０３は、バッファ１０４に転送するファイルの一部を格納するごとに

（ステップ４０２）、送信受付プログラム１０６を呼び出し、送信要求を送出する（ステップ４０３）。送信受付プログラム１０６を呼び出した結果がエラーであればデータの送信は失敗であり、ハードウェアのエラーまたはソフトウェアのエラーであるので、エラー処理を行い終了する（ステップ４０５）。バッファ１０４内のデータ送信を成功裡に終えると、現在処理中のファイルが全て転送済みであるか調べて未処理の領域があればそれらを処理するためにステップ４０２から繰り返す（ステップ４０６）。各ユーザプログラムは送信すべきファイルがまだある限り以上のステップを繰り返す（ステップ４０７）。このように各送信ユーザプログラム１０３は固定長のバッファ１０４を使いまわすようになっている。同様に各受信ユーザプログラム２０３も固定長のバッファ２０４を使いまわすようになっている。本発明を用いたデータ転送においては任意のサイズのデータを転送することが可能であるが、ユーザプログラムの持つバッファは固定長である必要がるので、固定長のバッファを用意し繰り返し使用している。従来技術を用いたデータ転送では、ユーザプログラムの持つバッファの大きさをどのように設定しても、データはＯＳが持つシステムバッファのサイズごとに分割されて転送されるので、分割処理あるいは統合処理のオーバヘッドが存在する。一方本発明を用いたデータ転送ではユーザプログラムの持つバッファ単位でデータ転送が行われ、ユーザプログラムによる分割処理あるいは統合処理は依然として存在するものの、ユーザプログラムの扱うデータの規模を考慮した適当なサイズのユーザバッファを利用することで、分割処理あるいは統合処理はユーザプログラムによる最小限の処理のみとなる。さらに、本実施例におけるファイル転送システムのような応用の場合では、直接ファイル入出力を行うため、データは分割処理あるいは統合処理は全く不要となる。各送信側ユーザプログラム１０３は、送信受付プログラム１０６を呼び出す際に、送信データが格納されているバッファ１０４の先頭アドレスと、バイト単位で示したファイルの大きさを表す送信サイズと、通信相手を特定するための通信路番号を引数として渡す。通信路番号は、特定の送信側ユーザプログラム１０３と特定の受信側ユーザプログラム２０３との組み合わせを識別するために用いるもので、システム内で一意な数値であり、ＴＣＰ／ＩＰにおけるコネクション識別子に相当する。通信路はＴＣＰ／ＩＰと同様に、通信を行う二つのユーザプログラムのそれぞれが明示的に通信路（コネクション）の終端（ソケット）を開設し、一方の終端に公知の（お互いが既知な）名前をつけ、もう一方の終端が名前付終端への接続を要求し、名前付終端が接続を許可することで成立する。通信路識別子（コネクション識別子）は二つの終端の組を一意に識別できるものであり、ユーザプログラムは複数個の通信路あるいは終端を作成し、あるいは使用できるが、本実施の形態においては二つのユーザプログラムの組が一つの通信路を扱っている。以下では、三つの送信側のユーザプログラム１０３の持つ通信路番号はそれぞれ１００１、１００２、１００３であると仮定する。また、三つの送信側のユーザプログラム１０３のそれぞれの通信相手となる受信側の三つのユーザプログラム２０３の持つ通信路番号も同一の通信路番号を持つ。送信サイズはゼロより大きくなければならない。

【００２６】（Ａ２）送信受付プログラム１０６による送信要求の受付

図４のフローチャートに示すように、送信受付プログラム１０６は、呼び出されると、まず新しい送信要求アイテム５１０（図５）を作成し（ステップ４１１）、その中の項目を埋める（ステップ４１２）。図５に示すように、送信要求アイテム５１０は、主記憶装置１０２の送信データを保持したバッファの先頭のアドレスを示す送信開始アドレス５１１と、どれだけのデータを送信するかを示す送信サイズ５１２と、送信先を特定する通信路番号５１３と、だれが送信したかを示す要求元プログラム識別子５１４と、送信要求アイテム５１０を連結するための次ポインタ５１５からなる。新しい送信要求アイテム５１０の送信開始アドレス５１１と送信サイズ５１２と通信路番号５１３にはユーザプログラム１０３からの呼び出し時に引数で与えられた情報を格納し、要求元プログラム識別子５１４には呼び出し元のユーザプログラム１０３のプログラム識別子を格納する。

【００２７】その後、送信受付プログラム１０６は、作成した送信要求アイテム５１０を送信要求リスト１０８に追加する（ステップ４１３）。すなわち、作成した送信要求アイテム５１０の次ポインタ５１５には、送信要求リスト１０８の中で最も新しい送信要求アイテム５１０の先頭アドレスを格納する。なお、新しい送信要求アイテム５１０を送信要求リスト１０８に追加する際には、送信要求リスト１０８内のそれぞれの送信要求アイテム５１０を調べて、その中に同一の通信路番号５１３を持ち、かつ送信サイズ５１２がゼロである送信要求アイテム５１０が存在するか否かを調べる。そのような送信要求アイテム５１０は後述する送信失敗の報告に用いられ、この時点ではすでに用済みであるので送信要求リスト１０８から削除する。

【００２８】その後、送信受付プログラム１０６は、現在実行中の処理を凍結して再び処理が解凍されるまで休止することで送信完了報告を待つ（ステップ４１４）。送信受付プログラム１０６はユーザプログラム１０３により呼び出されており実行中はユーザプログラム１０３のプログラム識別子を持つ。休止するのは「送信受付プログラム１０６を呼び出し中のユーザプログラム１０３」であり、あるユーザプログラム１０３が休止している最中に、別のユーザプログラム１０３が送信受付プログラム１０６を呼び出した場合には、送信受付プログラ

ム１０６は後者のユーザプログラム１０３のプログラム識別子を持って動作する。本実施の形態では、再開の指示は、後に説明する送信処理プログラム１０７が行う。

【００２９】（Ａ３）送信処理プログラム１０７（その１）ＲＥＱコマンドの送信

図６に示すように、送信処理プログラム１０７はまず、送信要求リスト１０８を定期的に調べ（ステップ４３１、４３２）、新しい送信要求アイテム５１０があればそれを取り出し送信要求リスト１０８から削除する（ステップ４３３）。その際に送信サイズ５１２がゼロである送信要求アイテム５１０は、後述の送信失敗報告のための送信要求アイテムであるので無視する。取り出した送信要求アイテム５１０の各々に対してＲＥＱコマンドを作成し、受信側の計算機２００に送る（ステップ４３４）。

【００３０】具体的には、送信処理プログラム１０７は、ステップ４３４では、複数の送信要求アイテムからＲＥＱコマンドブロック１０９を作成する。ＲＥＱコマンドブロック１０９は、図７（ａ）に示すように、それぞれ一つの送信要求アイテムに対して一つのＲＥＱコマンドを含む。各ＲＥＱコマンドは、コマンドコード５３１と転送サイズ５３２と通信路番号５３３を有する。本実施の形態での具体的な例として、送信側の三つのユーザプログラム１０３から新たに発行された三つのＲＥＱコマンドが、送信処理プログラム１０７でのステップ４３１、４３２の実行時に検出されたと仮定する。図７（ａ）に示すように、これらの三つのＲＥＱコマンドは、それぞれ通信路番号５３３として、１００１、１００２、１００３を有し、転送サイズ５３２として３００、３００、５００を有すると仮定する。

【００３１】送信処理プログラム１０７は、ＲＥＱコマンドブロック１０９を送信するために、さらにＤＭＡ制御リスト１１１に送信命令を書いて、通信装置１１２を起動する。ＤＭＡ制御リスト１１１は、図７（ｂ）に示すように、それぞれ一つの送受信命令に対応する複数のエントリからなり、各エントリは、送信命令か受信命令のいずれかであるかを示す送受信命令５０１と、送受信命令５０１が送信命令の場合に、送信すべきデータの主記憶装置１０２上の先頭アドレスを表し、あるいは送受信命令５０１が受信命令の場合に、受信すべきデータを格納すべき主記憶装置１０２上の先頭アドレスを表す送受信開始アドレス５０２と、送信データあるいは受信データのサイズを表す送受信サイズ５０３を含む。今の場合、ＲＥＱコマンドブロック１０９を送信するには、このＤＭＡ制御リスト１１１の一つのエントリとして、送受信命令５０１に送信命令を、送受信開始アドレス５０２にＲＥＱコマンドブロック１０９の先頭アドレスを、送受信サイズ５０３にはＲＥＱコマンドブロック１０９の大きさ（固定長）をそれぞれ格納する。

【００３２】（Ａ４）通信装置１１２（その１）（Ａ４１）ＲＥＱコマンドブロック１０９の送信

通信装置１１２は、起動されると、命令制御回路１１５が、ＤＭＡ制御リスト１１１の先頭のエントリを主記憶装置１０２から読み出し、今の場合には、このエントリには送信命令が含まれているので、送信動作を開始する。すなわち、この先頭のエントリに含まれた送受信開始アドレス５０２と送受信サイズ５０３に従い、ＲＥＱコマンドブロック１０９に含まれた複数のＲＥＱコマンドを連続して主記憶装置１０２から読み出すようＤＭＡ回路１１３に指示し、通信回路１１４に、この読み出されたＲＥＱコマンドブロック１０９をチャネル３００を介して直ちに計算機２００の通信装置２１２に送るよう指示する。命令制御回路１１５は、ＤＭＡ制御リスト１１１の後続のエントリに有効な命令があればそれに応答するが、今の場合にはＤＭＡ制御リスト１１１そのような後続の有効な命令はない。

【００３３】（Ａ４２）送信完了通知

命令制御回路１１５は、この送信を終えるとハードウェア割り込みを発生し、送信処理プログラム１０７に送信完了を通知する。送信処理プログラム１０７はこの割り込みを検知することで送信完了を確認する。

【００３４】（Ｂ）受信側計算機２００（その１）

（Ｂ１）受信側ユーザプログラム２０３（その１）受信要求の発行

本実施の形態では、送信側ユーザプログラム１０３がＯＳ１０５に対して送信要求を発行するとともに、受信側のユーザプログラム２０３がＯＳ２０５に対して受信要求を発行するように構成され、後に説明するように、送信要求を発行したユーザプログラム１０３からの送信データは、いずれかの送信先のユーザプログラム２０３がそのデータを受信可能な状態になった後に、具体的には、送信先のユーザプログラム２０３がそのデータに対して受信要求を発行した後にのみその送信先のユーザプログラムに転送される。

【００３５】図８のフローチャートに示すように、受信側ユーザプログラム２０３の処理も送信側ユーザプログラム１０３の処理と同様の２重のループ構造をなしており、処理の最初と最後に転送されたファイルを保持するための受信用のバッファ２０４を確保し、あるいは解放する（ステップ４５１、４５８）。外側のループは複数のファイルを処理するためのループであり、内側のループは各々のファイルを複数回の通信で転送するためのループである。

【００３６】各受信側ユーザプログラム２０３は、受信受付プログラム２０６を呼び出し、受信要求を通知し（ステップ４５２）、バッファ２０４に格納されたデータをファイルの一部として格納する（ステップ４５５）。

【００３７】受信受付プログラム２０６を呼び出しから戻ったときは、呼び出しからの戻り値を調べてエラーで

あることがわかるとエラー処理を行い終了する（ステップ４５４）。ファイルへの出力が済むと、現在処理中のファイルが全て転送済みであるか調べて未処理の領域があればそれらを処理するためにステップ４５２から繰り返す（ステップ４５６）。各ユーザプログラムは送信すべきファイルがまだある限り以上のステップを繰り返す（ステップ４５７）。

【００３８】（Ｂ２）受信受付プログラム２０６による受信要求の受付

図９のフローチャートに示すように、受信受付プログラム２０６は、呼び出されると、まず新しい受信要求アイテム５２０を作成し（ステップ４２１）、その中の項目を埋める（ステップ４２２）。受信要求アイテム５２０は、図１０に示すように、受信したデータを主記憶装置２０２のどのアドレス位置から格納するかを示す受信開始アドレス５２１と、どれだけのデータが受信可能かを示す最大受信サイズ５２２と、送信元のユーザプログラムとその受信要求を発行したユーザプログラムの組に対して定められた通信路番号５２３と、受信要求元のユーザプログラムの識別子５２４と、他の受信要求アイテム５２０と連結するための次ポインタ５２５からなる。作成した新しい受信要求アイテム５２０の受信開始アドレス５２１と最大受信サイズ５２２と通信路番号５２３には受信要求元のユーザプログラムからの呼び出し時に引数で与えられた情報を格納し、要求元プログラム識別子５２４には呼び出し元である受信側ユーザプログラム２０３のプログラム識別子を格納する。なお、最大受信サイズ５２２はゼロよりも大きくなければならない。

【００３９】受信受付プログラム２０６は、作成した受信要求アイテム５２０を受信要求リスト２０８に追加する（ステップ４２３）。すなわち、作成した受信要求アイテム５２０の次ポインタ５２５には、受信要求リスト２０８の中で最も新しい受信要求アイテム５２０のアドレスを格納する。なお、新しい受信要求アイテム５２０を受信要求リスト２０８に追加する際には、受信要求リスト２０８内のそれぞれの受信要求アイテム５２０を調べて、その中に同一の通信路番号５２３を持ち、かつ最大受信サイズ５２２がゼロである受信要求アイテム５２０が存在するか否かを調べる。そのような受信要求アイテム５２０は後述する受信失敗の報告に用いられ、この時点ではすでに用済みであるので受信要求リスト２０８から削除する。

【００４０】その後、受信受付プログラム２０６は、送信受付プログラム１０６と同様に現在実行中の処理を凍結して再び処理が解凍されるまで休止することで受信完了報告を待つ（ステップ４２４）。本実施の形態では、再開の指示は、次に説明する受信処理プログラム２０７が行う。

【００４１】（Ｂ３）受信処理プログラム２０７（その１）ＲＥＱコマンドブロックの受信命令

受信側の計算機２００では、送信側の計算機１００から送信されるＲＥＱコマンドブロックをいつでも受信できるように準備をしておく必要がある。受信処理プログラム２０７は、このコマンドブロックの受信を通信装置２１２に命令する。

【００４２】すなわち、図１１のフローチャートに示すように、受信処理プログラム２０７は、ＲＥＱコマンドブロックを受信し、主記憶装置２０２にＲＥＱコマンドブロック２０９として格納することを指示する受信命令をＤＭＡ制御リスト２１１に書いて、通信装置１１２を起動し、受信完了を待つ（ステップ４４１）。

【００４３】図１２（ｂ）に示すように、ＤＭＡ制御リスト２１１は、ＤＭＡ制御リスト１１１と同じ構造を有する。受信処理プログラム２０７は、ＲＥＱコマンドブロック１０９を受信してＲＥＱコマンドブロック２０９として主記憶装置２０２として格納するために、ＤＭＡ制御リスト２１１の一つのエントリに、送受信命令５０４として受信命令を、送受信開始アドレス５０５としてＲＥＱコマンドブロック２０９の先頭アドレスを、送受信サイズ５０６としてＲＥＱコマンドブロック２０９の大きさ（固定長）をそれぞれ格納する。

【００４４】（Ｂ４）通信装置２１２（その１）（Ｂ４１）ＲＥＱコマンドブロック１０９の受信

通信装置２１２が起動されると、命令制御回路２１５がＤＭＡ制御命令を解釈し、実行する。まず、命令制御回路２１５が、ＤＭＡ制御リスト２１１の最初のエントリを主記憶装置２０２から読み出し、このエントリ中の命令、今の場合には受信命令に従って通信回路２１４およびＤＭＡ回路２１３を制御する。すなわち、通信回路２１４にコマンドブロックの大きさのデータの受信を命じ、通信回路２１４がＲＥＱコマンドブロック１０９の内容をチャネル３００を介して計算機１００から受信すると、ＤＭＡ回路２１３に、受信したＲＥＱコマンドブロック１０９の内容を主記憶装置２０２上のＲＥＱコマンドブロック２０９に直ちに書き込ませる。すなわち、上記読み出したエントリに含まれた送受信開始アドレス５０５と送受信サイズ５０６に従い、受信したＲＥＱコマンドブロック１０９を主記憶装置２０２にＲＥＱコマンドブロック２０９として書き込む。図７（ｂ）には、三つの送信要求ＲＥＱを含むＲＥＱコマンドブロック２０９の例を示している。ＤＭＡ回路２１３は、ＤＭＡ制御リスト２１１の後続のエントリに有効な命令があればそれに応答するが、今の場合にはＤＭＡ制御リスト２１１そのような後続の有効な命令はない。

【００４５】（Ｂ４２）受信完了通知

命令制御回路２１５は、この書き込みが終了するとハードウェア割り込みを発生し、受信処理プログラム１０７に受信完了を通知する。受信処理プログラム１０７はこの割り込みを検知することにより受信完了を確認し、次のステップ４４１に進む。

【００４６】（Ｂ５）受信処理プログラム２０７（その２）

（Ｂ５１）ＡＣＫ／ＮＡＣＫコマンドの送信命令

ＲＥＱコマンドブロック２０９の受信が完了すると、受信処理プログラム２０７は、受信したＲＥＱコマンドブロック２０９内のそれぞれのＲＥＱコマンドの対して、受信が可能であるか否かを調べ、ＡＣＫコマンドまたはＮＡＣＫコマンドを作成し、それらのコマンドを含むＡＣＫコマンドブロックを送信側の計算機１００に送る（ステップ４４２）。

【００４７】すなわち、それぞれのＲＥＱコマンドが受信可能か否かを調べるために、受信要求リスト２０８から、ＲＥＱコマンドブロック２０９内のそれぞれの各ＲＥＱコマンドに対する通信路番号５３３と同じ通信路番号５２３を持つ受信要求アイテム５２０を探す。該当する受信要求アイテム５２０が存在しなければ、そのＲＥＱコマンドは受信不可能である。すなわち、ＲＥＱコマンドブロック２０９の受信が完了した時点で、その通信路番号５３３を指定する受信要求が受信側のユーザプログラム２０３のいずれかによりまだ発行されていない場合には、このＲＥＱコマンドは受信不可能となる。上記該当する受信要求アイテム５２０が存在した場合には、さらにそのＲＥＱコマンドに対する転送サイズ５３２がその該当する受信要求アイテム５２０の最大受信サイズ５２２より大きくないかを判別する。その転送サイズ５３２がその最大受信サイズ５２２を超えていればそのＲＥＱコマンドは受信不可能であるが、そうでなければそのＲＥＱコマンドは受信可能である。なお、このステップ４４２においては、最大受信サイズ５２２がゼロである受信要求アイテム５２０は、後述の受信失敗報告のための受信要求アイテムであるので無視する。受信不可能なＲＥＱコマンドには２種類あり、単に対応する受信要求が未発生の場合はＲＥＱコマンドの通知が早すぎただけなのでそのＲＥＱコマンドは送信側計算機１００において再び処理を試みられるが、サイズ超過の場合はプログラムミスなどの不測の事態であり、送信要求および受信要求は失敗としてそれぞれのユーザプログラムに通知される（この場合には受信要求はすでに発生している）。

【００４８】本実施の形態では、具体的な例示に当たり、送信側の三つのユーザプログラム１０３から図７（ａ）に例示する三つのＲＥＱコマンドが発行された後、ＲＥＱコマンドブロック２０９の受信が完了した時点までに、これらの三つのＲＥＱコマンドの内の二つに対応する受信要求が受信側の二つのユーザプログラム２０３から発行され、これら二つのＲＥＱコマンドのそれぞれが指定する転送サイズ５３２は、それらの対応する受信要求が指定する最大受信サイズを超えないと仮定する。さらに、上記三つのＲＥＱコマンドの内の第３のＲＥＱコマンドに対応する受信要求は受信側の他の一つのユーザプログラムからは遅れて発行されるが、その第３のＲＥＱコマンドが指定する転送サイズ５３２もその対応する第３の受信要求が指定する最大受信サイズを超えないと仮定する。この場合には、ＲＥＱコマンドブロック２０９の受信が完了した時点では、最初の二つのＲＥＱコマンドが受信可能となり、この最後のＲＥＱコマンドは受信不可能となる。

【００４９】こうして受信可能と判断されたＲＥＱコマンドおよび受信不可能と判断されたＲＥＱコマンドに対して、それぞれＡＣＫコマンドおよびＮＡＣＫコマンドを生成し、ＡＣＫコマンドブロック２１０として計算機１００に送る。図１２（ａ）に示すように、ＡＣＫコマンドブロック２１０には、ＲＥＱコマンドブロック２０９内の受信可能と判断されたＲＥＱコマンドに対しては、そのＲＥＱコマンドブロック２０９内のコマンド部５３１にあるＲＥＱを「ＡＣＫ」に変更しただけのＡＣＫコマンドが含まれる。さらに、ＲＥＱコマンドブロック２０９内の受信不可能と判断されたＲＥＱコマンドに対しては、そのＲＥＱコマンドブロック２０９内のコマンド部５３１にあるＲＥＱを「ＮＡＣＫ」に変更するとともに、受信不可能の原因が転送サイズの超過である場合は、その転送サイズ５３２を受信可能なサイズに変更する。送信側の計算機ではこの転送サイズ５３２の変更の有無を調べることによって、受信不能の理由が受信の未準備によるものかサイズ超過によるもの（致命的エラー）かを判断できる。

【００５０】図１２（ａ）の例では、ＡＣＫコマンドブロック２１０は、二つのＡＣＫコマンドと一つのＮＡＣＫコマンドからなる。すなわち、通信路番号が１００１および１００２であるＲＥＱコマンドは、対応する受信要求がすでに発生しており、要求される転送サイズも受信可能な量を超えていないので、受信可能でありＡＣＫコマンドが返される。一方、通信路番号が１００３であるＲＥＱコマンドは、対応する受信要求が未発生なために受信不可でありＮＡＣＫコマンドが返される。この受信不可の原因は転送サイズ超過ではなく受信要求未発生であるので、ＮＡＣＫコマンドの転送サイズ５３２は変更されない。

【００５１】受信処理プログラム２０７は、生成したＡＣＫコマンドブロック２１０を計算機１００に送信する命令を発行する。その命令は、計算機１００におけるＲＥＱコマンドブロック１０９の送信と同様である。すなわち、受信処理プログラム２０７は、ＤＭＡ制御リスト２１１内に、送受信命令５０４として送信命令を、送受信開始アドレス５０５としてＡＣＫコマンドブロック２１０の先頭アドレスを、送受信サイズ５０６としてＡＣＫコマンドブロック２１０の大きさ（固定長）をそれぞれ有する送信命令を格納し、通信装置２１２を起動する。その後、送信完了を確認するために、通信装置２１２からのハードウェア割り込みを待つ。

【００５２】（Ｂ５２）受信失敗報告
以上の仮定的な例では発生していないが、あるＲＥＱコマンドに対して通信路番号が一致する受信要求がすでに受信側のいずれかのユーザプログラムからすでに発行されていながら、そのＲＥＱコマンドが指定する転送サイズがその受信要求が指定する最大受信サイズを超過した場合、受信処理プログラム２０７は、その受信要求を発行した受信側ユーザプログラム２０３に受信失敗を報告する。すなわち、受信処理プログラム２０７は、受信に失敗した受信要求アイテム５２０の最大受信サイズ５２２をゼロに変更し、変更後の受信要求アイテム５２０を受信要求リスト２０８に戻し、さらにその受信要求アイテム５２０の要求元プログラム識別子５２４で特定される受信完了待ちの受信側ユーザプログラム２０３を再起動する（ステップ４４４（図１１））。
【００５３】受信受付プログラム２０６は、（ユーザプログラム２０３の呼び出し延長処理として）処理を再開されると、受信要求リスト２０８内に自分のプログラム識別子（この場合は呼び出し元のユーザプログラム２０３のプログラム識別子）を持った受信要求アイテム５２０が存在するかどうかを調べて、受信側ユーザプログラム２０３に返る。返り値には、受信要求が戻されていれば失敗を表す返り値を、なければ成功を表す返り値を用いる（ステップ４２５（図９））。
【００５４】（Ｂ６）ユーザプログラム２０３（その２）
受信受付プログラム２０６が終了すると呼び出し元のユーザプログラム２０３に処理が戻る。ユーザプログラム２０３は呼び出しの結果の戻り値を調べて（ステップ４５３（図８））、エラーであればエラー処理を行い終了する（ステップ４５４（図８））。
【００５５】以上の仮定的な例では呼び出しは成功裡に終わりエラー処理は起こっていない。
【００５６】（Ｃ）送信側計算機（その２）
（Ｃ１）送信処理プログラム１０７（その２）ＡＣＫコマンドブロックの受信命令
送信側の計算機１００ではＡＣＫ／ＮＡＣＫコマンドブロック２１０を受け取る準備を行う。すなわち、図８のステップ４３５に示すように、送信処理プログラム１０７は、ＤＭＡ制御リスト１１１に、ＡＣＫコマンドブロックの受信命令を書いて、通信装置１１２を起動し、受信完了を待つ。この命令は、ＡＣＫコマンドブロックを受信し、主記憶装置１０２にＡＣＫコマンドブロック１１０として書き込むことを要求する。具体的には、送信処理プログラム１０７は、ＤＭＡ制御リスト１１１に、送受信命令５０１に受信命令を、送受信開始アドレス５０２にＡＣＫコマンドブロック１１０の先頭アドレスを、送受信サイズ５０３にはＡＣＫコマンドブロック１１０の大きさ（固定長）をそれぞれ格納する。受信完了の確認は、通信装置１１２からの割り込みを待つことで行う。
【００５７】（Ｃ２）通信装置１１２
通信装置１１２は、この受信命令に応答して、ＡＣＫコマンドブロック２１０を受信し、主記憶装置１０２にＡＣＫコマンドブロック１１０として書き込む。このときのこの回路の動作は、通信装置２１２がＲＥＱコマンドブロックを受信した場合と同じである。
【００５８】（Ｃ３）送信処理プログラム１０７（その３）データ送信命令
送信側の計算機１００は、ＡＣＫコマンドブロックを受け取るとデータの送信を開始する。すなわち、送信処理プログラム１０７は、図８のステップ４３６において、ＡＣＫコマンドブロック１１０に含まれた複数のＡＣＫコマンドをそれぞれ取り出し、それぞれに対応する送信データの送信を通信装置１１２に命令し、これを起動し、そのデータの送信の完了を待つ。各ＡＣＫコマンドに対する送信データは、そのＡＣＫコマンドと同じ通信路番号を有する送信要求アイテム５１０を送信要求リスト１０８から検索し、その送信要求アイテム５１０内の送信開始アドレス５１１、送信サイズ５１２でもって指定する。すなわち、送信処理プログラム１０７は、ＤＭＡ制御リスト１１１内の送受信命令５０１に送信命令を、送受信開始アドレス５０２に検索された送信開始アドレス５１１を、送受信サイズ５０３に検索された送信サイズ５１２をそれぞれ格納することでもって、そのＡＣＫコマンドに対するデータの送信を通信装置１１２に命令する。
【００５９】送信処理プログラム１０７は、ＡＣＫコマンドブロック１１０に複数のＡＣＫコマンドがある場合には、それらに対応する複数の送信命令をＤＭＡ制御リスト１１１に格納する。それらの送信命令をＤＭＡ制御リスト１１１に格納する順序は受信されたＡＣＫコマンドブロック１１０内に格納されていた対応するＡＣＫコマンドの順序と同じにする。
【００６０】送信処理プログラム１０７は、受信されたＡＣＫコマンドブロック１１０内にＮＡＣＫコマンドがある場合には、それぞれのＮＡＣＫコマンドに対して２通りの処理のいずれか一方を行う。すなわち、各ＮＡＣＫコマンド内の転送サイズ５３２の値が、対応するＲＥＱコマンドが要求した送信サイズ、すなわち、そのＲＥＱコマンドに対応する送信要求アイテム５１０内の送信サイズ５１２に等しければ、受信不可の原因は受信側計算機２００で受信要求が未だ発生していないことにある。従って、送信処理プログラム１０７は、ＲＥＱコマンドを再送するために、送信要求アイテム５１０を送信要求リスト１０８に戻す。
【００６１】一方、そのＮＡＣＫコマンド内の転送サイズ５３２が対応するＲＥＱコマンドが要求した送信サイズより縮小されていれば、受信不可の原因は、対応するＲＥＱコマンドが要求した送信サイズが、受信側の計算

機２００で発生した受信要求が指定した最大受信サイズを超過しているためである。この場合は、送信処理プログラム１０７は送信側ユーザプログラム１０３に送信失敗の報告を行う。すなわち、送信処理プログラム１０７は、失敗した送信要求アイテム５１０の要求元プログラム識別子５１４で特定される送信完了待ちの送信側ユーザプログラム１０３の処理を再開する。その際に、失敗した送信要求アイテム５１０の送信サイズ５１２をゼロに変更し、送信要求リスト１０８に戻す。

【００６２】（Ｃ４）通信装置１１２（その３）データ送信

通信装置１１２は起動されると、送信処理プログラム１０７により命令されたデータの送信をＤＭＡ制御リスト１１１に基づいて行う。すなわち、ＤＭＡ回路１１３は、ＤＭＡ制御リスト１１１に記載された各送信命令が指定する送信データをユーザプログラム１０３内のバッファ１０４からＤＭＡ回路１１３が読み出し、通信回路１１４が送信し、通信回路１１４は、送信が完了すると割り込みにより送信処理プログラム１０７にそれを通知する。命令制御回路１１５はＤＭＡ制御リスト１１１に複数のデータの送信命令が記載されているときには、それらの命令の記載順に従ってそれらの送信命令が指定する複数の送信データを順次読み出し、通信回路１１４がそれらのデータを順次送信する。このように、送信側では、送信データはユーザプログラム１０３のバッファからＯＳ１０５にコピーされることなく送信される。また、ユーザプログラム１０３内の送信データは分割されることなく、連続して送信される。今の例の場合、二つのデータを送信し、二つの送信を終えるとハードウェア割り込みを発生させる。

【００６３】（Ｃ５）送信完了報告

送信処理プログラム１０７は、データ送信の完了を通信装置１１２から通知されると、送信側ユーザプログラム１０３に送信の完了報告を行う（ステップ４３７（図６））。図１２（ａ）に示されたＡＣＫコマンドブロック２０９を送信した後では、二つのＡＣＫコマンドに対する二つのユーザプログラム２０３に送信完了の報告を行う。この際、送信の失敗報告と異なり、対応する送信要求アイテム５１０を送信要求リスト１０８に戻さない。

【００６４】（Ｃ６）ユーザプログラム１０３（その２）送信失敗に対する処理

送信側のユーザプログラム１０３は送信受付プログラム１０６の呼び出しから返ってくると、その戻り値を調べ（ステップ４０４（図３））、エラーであればエラー処理を行う（ステップ４０５（図３））。

【００６５】（Ｄ）受信側計算機２００（その２）

（Ｄ１）受信処理プログラム２０７（その３）データ受信命令

受信処理プログラム２０７は、ＡＣＫコマンドブロックの送信命令を発行後、そのＡＣＫコマンドブロックによりＡＣＫコマンドを一つでも返信していれば（ステップ４４３（図９））、そのＡＣＫコマンドに対応するデータの送信命令の送信を通信装置２１２に命令し、通信装置２１２を起動し、その送信の完了を待つ（ステップ４４４（図９））。ＡＣＫコマンドに対応して受信すべきデータは、そのＡＣＫコマンドに含まれる通信路番号を有する受信要求アイテム５２０を受信要求リスト２０８から検索し、その受信要求アイテム５２０内の受信開始アドレス５２１で指定されるデータである。従って、送信処理プログラム１０７は、ＤＭＡ制御リスト２１１内の送受信命令コード５０４に受信命令コードを、送受信開始アドレス５０５に検索された受信開始アドレス５２１を、送受信サイズ５０６に返したＡＣＫコマンドで指定した転送サイズ５３２をそれぞれ格納することでもって、そのＡＣＫコマンドに対するデータの受信を通信装置２１２に命令する。そのＡＣＫコマンドブロック２１０に複数のＡＣＫコマンドが格納されていた場合には、それらのＡＣＫコマンドの順序に従って、それぞれに対応するデータ受信命令をＤＭＡ制御リスト２１１に格納する。

【００６６】なお、送信処理プログラム１０７は、ステップ４４３の判定の結果、送信したＡＣＫコマンドブロック２１０によりＡＣＫコマンドを一つも返さなかったと判定されたときには、データ受信命令の発行は行わずにステップ４４１に戻り、もう一度ＲＥＱコマンドブロックを受けるための命令を発行する。

【００６７】（Ｄ２）通信装置２１１（その３）

（Ｄ２１）データの受信

通信装置２１２は起動されると、受信処理プログラム２０７により命令されたデータの受信をＤＭＡ制御リスト２１１に基づいて行う。すなわち、命令制御回路２１５は、ＤＭＡ制御リスト２１１の先頭のエントリを主記憶装置２０２から読み出し、通信回路２１４がデータを送信側計算機１００から受信すると、読み出したエントリに含まれた受信命令が指定する、いずれかのユーザプログラム１０３内のバッファ１０４にその受信されたデータを書き込む。ＤＭＡ制御リスト２１１に複数のデータの受信命令が記載されているときには、命令制御回路２１５は、それらの命令の記載順に従って、順次受信された複数のデータをそれらの受信命令が指定する、複数のユーザプログラム２０３内のバッファ２０４に順次書き込む。図１２（ａ）に示されたＡＣＫコマンドブロックを送信した後では、二つのＡＣＫコマンドに対する二つのデータが受信される。このように、受信側でも、通信装置２１４により受信されたデータは、従来技術のようにＯＳ２０５内のバッファにコピーされることなく、直接いずれかのユーザプログラム２０３のバッファ２０４に書き込まれる。

【００６８】（Ｄ２２）受信完了通知

命令制御回路２１５は、いずれかの受信データの書き込みを完了すると、ハードウェア割り込みを発生し、受信処理プログラム２０７にデータ受信の完了を通知する。受信処理プログラム１０７はこの割り込みを検知することにより受信完了を確認し、次のステップ４４５に進む。なお、ＤＭＡ制御リスト２１１に複数のデータの受信命令が記載されているときには、それらの命令が指定する複数のデータの各々をいずれかのバッファ２０３に書き込むごとに、上記受信完了割り込みを発生する。

【００６９】（Ｄ３）受信処理プログラム２０７（その４）

（Ｄ３１）受信完了報告

通信装置２１１によりデータ受信完了の通知がなされると、受信処理プログラム２０７は、受信したデータに対応するＡＣＫコマンドに対応する受信要求元のユーザプログラム２０３に受信の完了報告を行う（ステップ４４５（図９））。図１２（ａ）に示されたＡＣＫコマンドブロックを送信した後では、二つのＡＣＫコマンドに対する二つのユーザプログラム２０４に受信完了の報告を行う。この際、対応する受信要求アイテム５１０を受信要求リスト１０８に戻さない。

【００７０】（Ｄ３２）受信側ユーザプログラム２０３

受信側のユーザプログラム２０３は受信受付プログラム２０６の呼び出しから戻ると戻り値を調べる。失敗の場合はＡＣＫコマンドブロック２１０を作成する時点で戻って来、この例のこの時点では必ず成功裡に戻ってくる。戻ってくるとバッファ２０４に受信したデータをファイルの一部に出力し（ステップ４５５（図８））、出力が済むと、現在処理中のファイルが全て転送済みであるか調べて未処理の領域があればそれらを処理するためにステップ４５２から繰り返す（ステップ４５６）。各ユーザプログラムは送信すべきファイルがまだある限り以上のステップを繰り返す（ステップ４５７）。

【００７１】（Ｅ）ＮＡＣＫコマンドに対するデータの転送

今考えている仮定的な例では、通信路番号が１００３である送信要求アイテム５１０に対しては、対応する受信要求が受信側の計算機２００で出力されなかったためにＮＡＣＫコマンドがそこから返送された。このため、この送信要求アイテム５１０に対する送信データは送信されなかったが、このデータは以下のようにしてその後に送信される。

【００７２】図８に示すように、送信処理プログラム１０７は、受信したＡＣＫコマンドに対して前述した送信命令を発行した後、ステップ４３２へ処理を移して新たな送信要求を探す。前述したデータを送信しなかった送信要求アイテム５１０は送信処理プログラム１０７により、送信要求リスト１０８に戻されている。従って、送信処理プログラム１０７が新たな送信要求を探すした時点で、この送信要求アイテム５１０が探索され、先に記載した方法により対応するＲＥＱコマンドを含むＲＥＱコマンドブロックが計算機２００に送られる。その時点で、受信側の計算機２００で、通信路番号が１００３である受信要求がすでに発行されているときには、受信側の計算機２００からこのＲＥＱコマンドに対するＡＣＫコマンドを含むＡＣＫコマンドブロックが先に記載した方法により返送され、その後、このＲＥＱコマンドに対する送信データが計算機１００から送信され、計算機２００により受信される。こうして、先に送信失敗した送信データも送信することができる。

【００７３】以上のように、本実施の形態によれば、ユーザプログラムのバッファとＯＳのバッファの間のデータコピー処理が送信側計算機と受信側計算機のいずれにおいても必要でないため、計算機間データ転送を高速に行える。

【００７４】従来のＴＣＰ／ＩＰによるデータ転送では、受信側ユーザプログラムの準備が整っていなくてもデータは送信される。その場合には、受信側の計算機において一旦受信データをＯＳバッファに蓄えてから受信側ユーザプログラムのバッファにコピーする必要があり、このデータコピー処理のオーバヘッドが高速なデータ転送の実現を妨げていた。また、受信準備が整っていない受信側ユーザプログラムがあまりに多いとＯＳバッファが溢れてデータを再送しなければならず、この再送に備えて送信側の計算機でもＯＳバッファに一旦データを蓄えておく必要があった。このため送信側の計算機でもデータコピーが発生し、そのオーバヘッドが高速なデータ転送の実現を妨げていた。

【００７５】しかるに、本実施の形態によるデータ転送ではデータ転送に先立つコマンドのやりとりで受信準備の完了を確認し、受信準備が整っていない場合にはデータそのものでなくコマンドのみを再送する方法を採ることにより、従来のデータコピーを不要にしている。

【００７６】また、本実施の形態では、ＯＳの固定長の通信バッファの大きさを超える大きなデータを転送する場合にもデータを分割して転送する必要がないため、通信処理のオーバヘッドが小さく、データ転送を高速に行える。

【００７７】従来のＴＣＰ／ＩＰによるデータ転送では、事前に送受信するデータサイズを調停する手段を持たないため、転送サイズはＯＳの設定した固定値である必要がある。従って、これを超える大きなデータを転送する場合に、送信側のＯＳで転送サイズごとにデータを分割しておき、受信側のＯＳでは受けたデータを再構成する必要がある。このデータの分割、再構成処理のオーバヘッドのために高速なデータ転送が実現できなかった。

【００７８】しかるに、本実施の形態によるデータ転送では事前に送受信するデータサイズを調停することで任意の大きさのデータを転送できるので、データの分割、

統合処理は不要となり、高速なデータ転送が実現可能である。同様に通信装置が転送するデータの個数も事前に調停することで複数のデータ転送の通信を一括して処理できるので、通信装置の起動処理、後処理（割り込み処理）が削減され、さらに高速なデータ転送が実現可能である。

【００７９】従来のＨＤＬＣによるデータ転送では、本発明と同様にデータ転送に先立ってデータ転送の調停を行うので、データコピーのオーバヘッドは存在しない。しかし、データ転送にＯＳが介在せず、転送を行うユーザプログラムは通信装置を独占してしまい、同時に複数のユーザプログラムがデータ転送を行えないという欠点があった。

【００８０】しかるに、本実施の形態によるデータ転送では、調停作業をＯＳが一括して行うために、データコピーのオーバヘッドがないという利点を保ったまま、同時に複数のデータ転送の調停を矛盾なく行うことができる。

【００８１】＜変形例＞本発明は、以上の実施の形態に限定されるものではなく、いろいろの形態で実施可能である。例えば、

（１）　上に示した発明の実施の形態では、二つの計算機のみが接続されているシステムであったが、より多くの計算機が相互に接続されているシステムでもよい。この場合、異なる計算機対に対応した伝送路が各計算機対に対して設けられ、各計算機対がその対応する伝送路により接続されていることが望ましい。

【００８２】（２）　計算機を接続する伝送路としては、実施の形態で示したチャネルに限らず、例えば、複数の計算機を接続するインタコネクトネットワークでもよい。例えば、クロスバスイッチ、ハイパクロスバスイッチあるいは多段スイッチネットワークでもよい。

【００８３】（３）　これらの計算機間でのデータ転送は、パケットの形でデータを転送するものでもよい。例えば、上記ハイパクロスバネットワークで接続された複数の計算機間では通常はデータはパケットの形式で転送される。各パケットに転送先の計算機を指定するアドレスと転送すべきデータその他が含まれる。本発明は、このような形式でデータを計算機間で転送するシステムにも適用できる。

【００８４】（４）　実施の形態では、ユーザプログラムとＯＳ間の入出力インタフェースは同期型、すなわちユーザプグラムはＯＳに依頼した入出力処理が完了するまで待つタイプの入出力インタフェースであったが、非同期型の入出力インタフェースでもよい。実施の形態で示したファイル転送システムを非同期入出力インタフェースを備えたファイル転送システムにするには、送受信の受付プログラムが送受信の完了を待たずに直ちにユーザプログラムに制御を戻し、ユーザプログラムは送受信の要求を発行した後、改めて送受信の完了を待つ要求を受付プログラムに発行し、受付プログラムはこれを受けて残りの処理を行えばよい。

【００８５】

【発明の効果】以上の記載から明らかなように、本発明によれば、ユーザプログラムのバッファとＯＳのバッファの間のデータコピー処理がないため、計算機間のデータ転送を高速に行える。

【００８６】さらに、大きなデータを転送する場合にもデータを分割して転送する必要がないため、通信処理のオーバヘッドを小さくでき、データ転送を高速に行える。

【図面の簡単な説明】

【図１】本発明に係るデータ転送方法を実施する計算機システムの全体構成図。

【図２】図１の装置におけるデータ転送のタイムチャート。

【図３】図１の装置に使用する送信側ユーザプログラム（１０３）のフローチャート。

【図４】図１の装置に使用する送信受付プログラム（１０６）のフローチャート。

【図５】図１の装置に使用する送信要求リスト（１０８）のデータ構造を示す図。

【図６】図１の装置に使用する送信処理プログラム（１０７）のフローチャート。

【図７】（ａ）は、図１の装置に使用するＲＥＱコマンドブロック（１０９）のデータ構造を示す図。（ｂ）は、図１の装置に使用するＤＭＡ制御リスト（１１１）のデータ構造を示す図。

【図８】図１の装置に使用する受信側ユーザプログラム（２０３）のフローチャート。

【図９】図１の装置に使用する受信受付プログラム（２０６）のフローチャート。

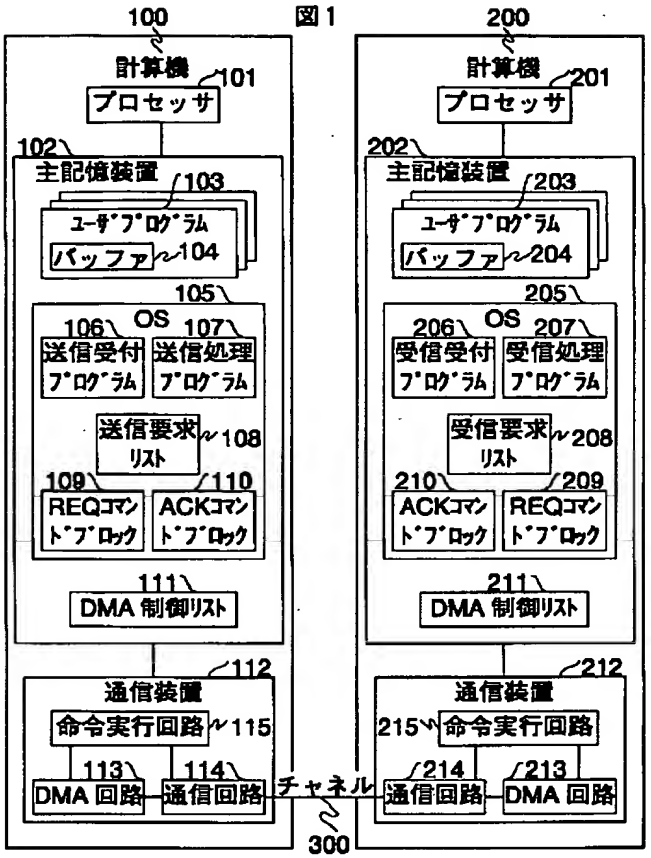
【図１０】図１の装置に使用する受信要求リスト（２０８）のデータ構造を示す図。

【図１１】図１の装置に使用する受信処理プログラム（２０７）のフローチャート。

【図１２】（ａ）は、図１の装置に使用するＡＣＫコマンドブロック（２１０）のデータ構造を示す図。（ｂ）は、図１の装置に使用するＤＭＡ制御リスト（２１１）のデータ構造を示す図。

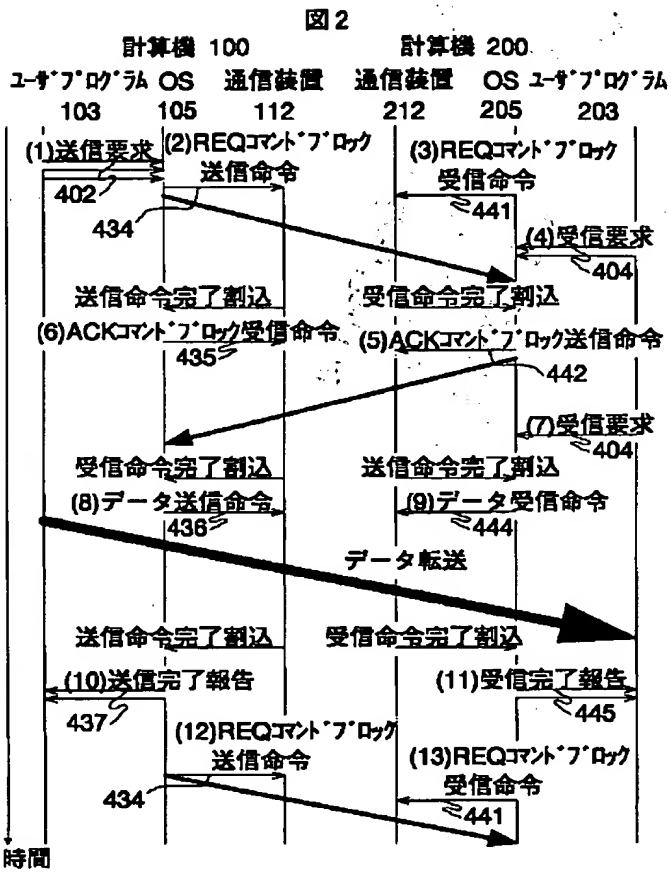
【図１】

図１



【図２】

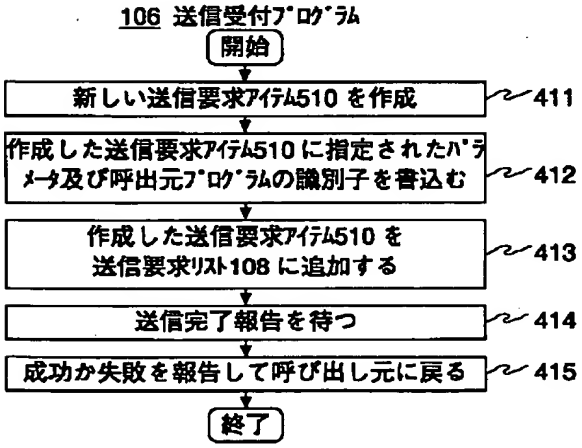
図２



【図４】

図４

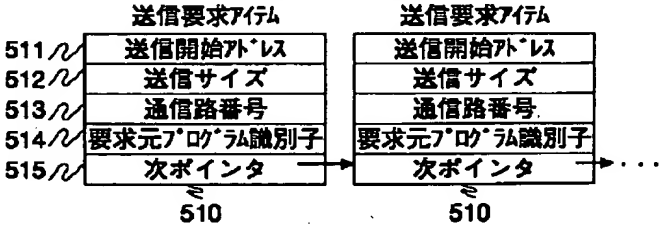
106 送信受付プログラム



【図５】

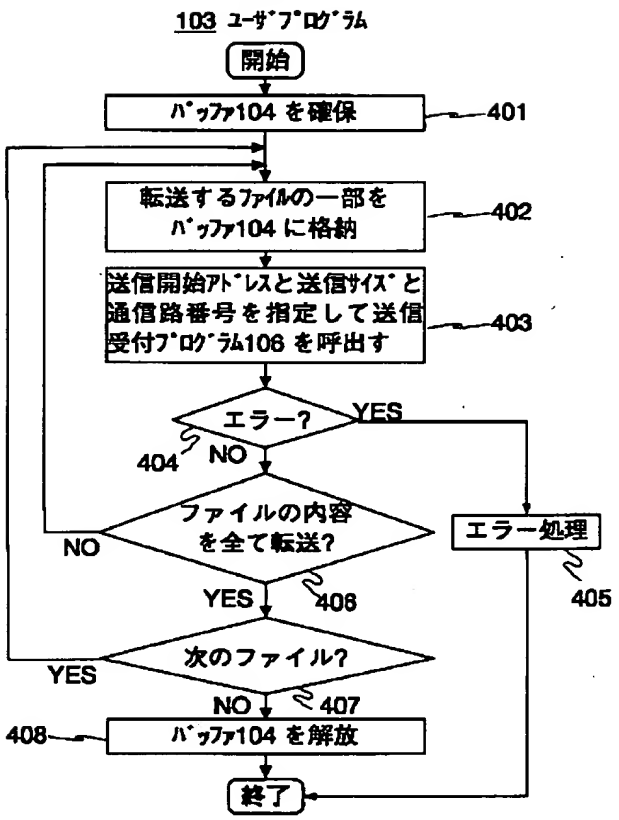
図５

108 送信要求リスト



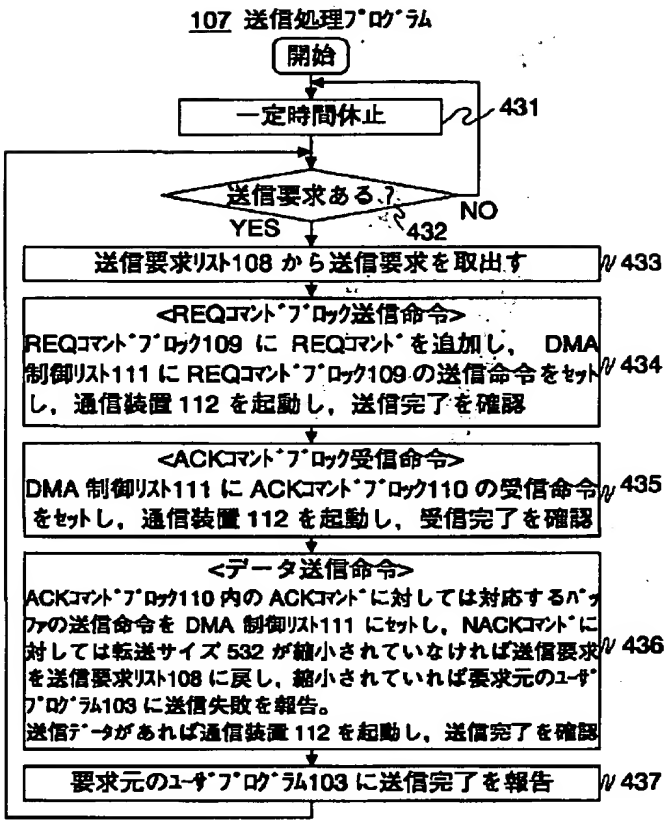
【図３】

図３



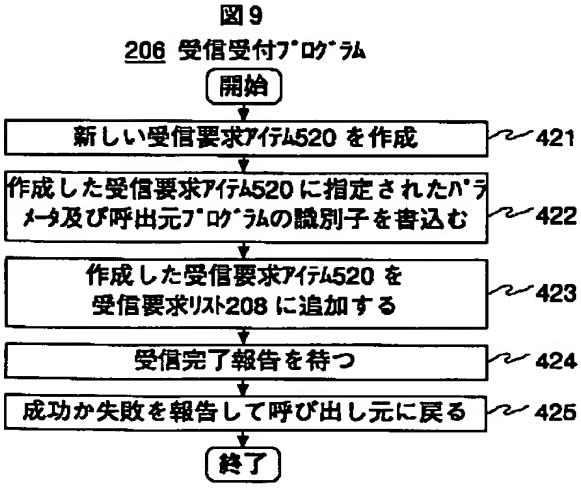
【図６】

図６



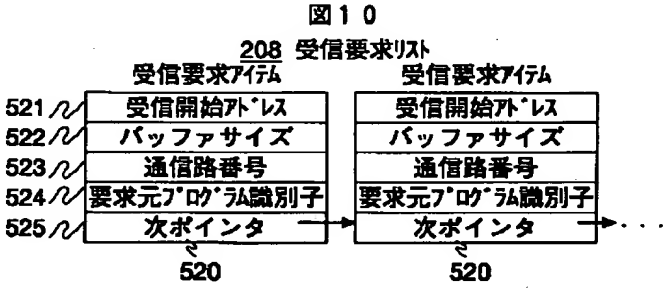
【図９】

図９



【図１０】

図１０



【図７】

図７

(a)

109 REQｺﾏﾝﾄﾞﾌﾞﾛｯｸ

ｺﾏﾝﾄﾞｺｰﾄﾞ	転送ｻｲｽﾞ	通信路番号
REQ	300	1001
REQ	300	1002
REQ	500	1003

531 532 533

(b)

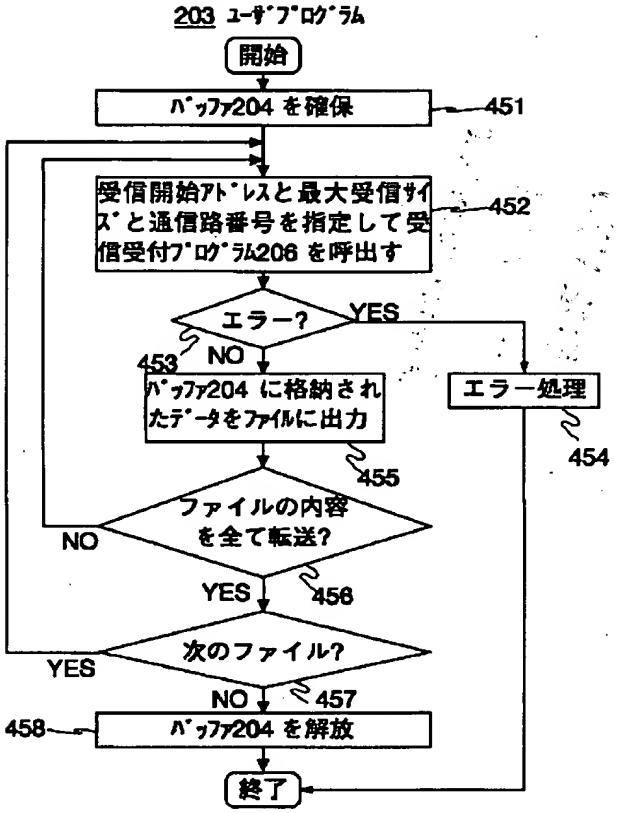
111 DMA 制御リスト

命令コード	送受信開始ｱﾄﾞﾚｽ	送受信ｻｲｽﾞ
SEND	A	a

501 502 503

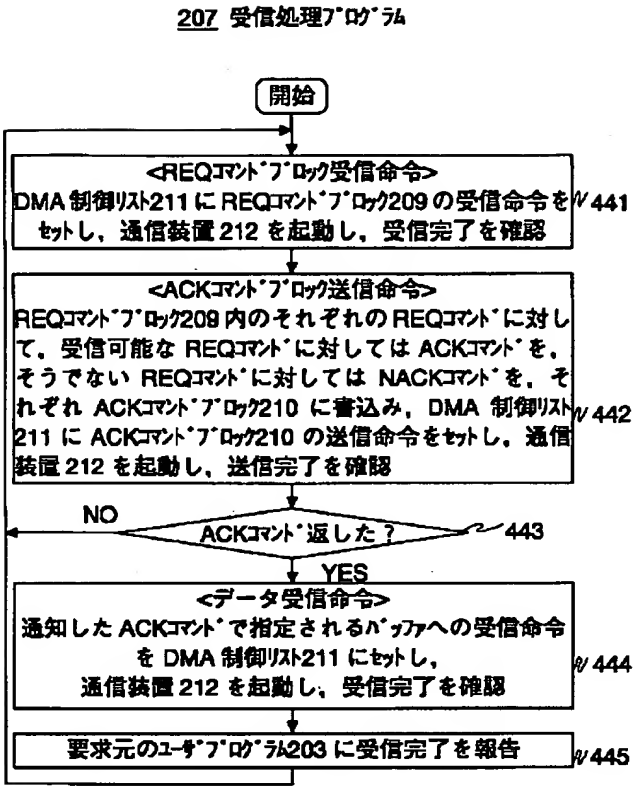
【図８】

図８



【図１１】

図１１



【図１２】

図１２

(a)

210 ACKｺﾏﾝﾄﾞﾌﾞﾛｯｸ

ｺﾏﾝﾄﾞｺｰﾄﾞ	転送ｻｲｽﾞ	通信路番号
ACK	300	1001
ACK	300	1002
NACK	500	1003

531　532　533

(b)

211 DMA制御リスト

命令ｺｰﾄﾞ	送受信開始ｱﾄﾞﾚｽ	送受信ｻｲｽﾞ
SEND	B	b

504　505　506

フロントページの続き

(72)発明者　清水　正明
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内

(72)発明者　鍵政　豊彦
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内